# Greenwich Apparent Sidereal Time

This ***Numerit*** program (`demogast`) demonstrates how to calculate the Greenwich apparent sidereal time using both a low and high precision algorithm. Both numerical methods are based on the IAU 1980 nutation algorithm.

## **`function gast1`**

This function calculates the apparent Greenwich sidereal time using the first few terms of the IAU 1980 nutation algorithm.

The Greenwich apparent sidereal time is given by the expression

$$q = q_m + \Delta y \cos\left(e_m + \Delta e\right) \tag{1}$$

In this equation $q_m$ is the Greenwich mean sidereal time, $\Delta y$ is the nutation in longitude, $e_m$ is the mean obliquity of the ecliptic and $\Delta e$ is the nutation in obliquity.

The Greenwich *mean* sidereal time, in degrees, is calculated using the expression

$$q_m = 100.46061837 + 36000.770053608T + 0.000387933T^2 - T^3/38710000 \tag{2}$$

where $T = \left(JD - 2451545\right)/36525$ and $JD$ is the Julian date on the UT1 time scale.

The mean obliquity of the ecliptic is determined from

$$e_m = 23°26'21.''448 - 46.''8150T - 0.''00059T^2 + 0.''001813T^3 \tag{3}$$

The nutations in obliquity and longitude involve the following three trigonometric arguments (in degrees):

$$\begin{aligned} L &= 280.4665 + 36000.7698T \\ L' &= 218.3165 + 481267.8813T \\ \Omega &= 125.04452 - 1934.136261T \end{aligned} \tag{4}$$

The calculation of the nutations use the following two equations:

$$\begin{aligned} \Delta y &= -17.20\sin\Omega - 1.32\sin 2L - 0.23\sin 2L' + 0.21\sin 2\Omega \\ \Delta e &= 9.20\cos\Omega + 0.57\cos 2L + 0.10\cos 2L' - 0.09\cos 2\Omega \end{aligned} \tag{5}$$

where these corrections are in units of arc seconds.

The syntax of this function is as follows:

```
function gast1 (jdate, gst)

` Greenwich apparent sidereal time

` input

`  jdate = Julian date

` output

`  gst = Greenwich apparent sidereal time (radians)
`        (0 <= gst <= 2 pi)
```

### **function gast2**

This function calculates the mean or apparent Greenwich sidereal time. For the apparent sidereal time calculation, the obliquity in longitude and obliquity are determined using a ***Numerit*** function named `nutation` which implements the full IAU 1980 nutation algorithm (108 terms). Please note that the Julian date can be passed to this function in a high-order (integer) and low-order (fractional) part. In the argument list `k` determines the type of Greenwich sidereal time calculation (`k = 0` $\Rightarrow$ mean, `k = 1` $\Rightarrow$ apparent). This function was ported to ***Numerit*** using the Fortran version of the NOVAS (Naval Observatory Vector Astrometry Subroutines) source code which was developed at the United States Naval Observatory.

For this numerical method, the nutation in longitude is determined from a trigonometric series of the form

$$\Delta y = \sum_{i=1}^{n} S_i \sin A_i \tag{6}$$

The nutation in obliquity is determined from a series of the form

$$\Delta e = \sum_{i=1}^{n} C_i \cos A_i \tag{7}$$

where

$$A_i = a_i\, l + b_i\, l' + c_i\, F + d_i\, D + e_i\, \Omega$$

In this expression $l, l', F, D$ and $\Omega$ are fundamental arguments.

The syntax for this ***Numerit*** function is as follows:

```
function gast2 (tjdh, tjdl, k, gst)

` this function computes the greenwich sidereal time
` (either mean or apparent) at julian date tjdh + tjdl

` input

`  tjdh = julian date, high-order part

`  tjdl = julian date, low-order part

`         julian date may be split at any point, but for
`         highest precision, set tjdh to be the integral part of
`         the julian date, and set tjdl to be the fractional part

`  k    = time selection code
`         set k=0 for greenwich mean sidereal time
`         set k=1 for greenwich apparent sidereal time

` output

`  gst = greenwich (mean or apparent) sidereal time in hours
```

The program will prompt you for the calendar date and universal time. The following is a typical draft output created with this program:

```
program demogast - Greenwich apparent sidereal time

calendar date    October 21, 1999

universal time   10 h 20 m 30 s

Julian date      2451472.930902778

low precision    12 h 18 m 11.32 s

high precision   12 h 18 m 11.31 s
```